A Specification for Defining and Annotating Regions of Macromolecular Structures

Steven E. Brenner
S.E.Brenner@bioc.cam.ac.uk
MRC Laboratory of Molecular Biology
Hills Road, Cambridge CB2 2QH, England
Telephone: +44 1223 248011, Fax: +44 1223 213556

Tim J. P. Hubbard
th@mrc-lmb.cam.ac.uk
Cambridge Centre for Protein Engineering
Hills Road, Cambridge CB2 2QH, England
Telephone: +44 1223 402131, Fax +44 1223 402140

Abstract

We present a program- and machine-independent standard for annotating macromolecular structures. Data encoded by this specification may be used for communicating information about structures and for exchanging it between different computer systems. The format consists of a set of ASN.1 objects which are mechanically straightforward to parse, but are also easy for humans to create and understand. It differs from all other related standards in that it specifies how a molecule should be displayed without requiring a custom format for the coordinate data.

Introduction

Biological macromolecular structure information has recently become widely available to researchers without specialized equipment or software. The Protein Data Bank (PDB) (Abola *et al.* 1987), the current repository for protein and, to a lesser extent, DNA structures now has a World Wide Web (WWW) site (Skora 1994) and NIH has developed a system for automatically manipulating structures, generating images, and sending them over the network (FitzGerald 1994).

Because the equipment and software necessary to view protein structures has also become commonplace (Hall 1995), protein structure coordinates—rather than simply 2D images—are now being used as a means of scientific communication, as exemplified by *Protein Science*'s inclusion of kinemages with every issue (Neurath 1992). Kinemages (Richardson & Richardson 1992; White *et al.* 1994) do more than just present the protein structure: they highlight various regions using color and other display techniques, provide preferred views, and also permit the display of special interactions, structural changes, and other dynamic features.

New databases, including scop (Murzin *et al.*, 1995), make use of the easy network access to protein structures to enhance their usability and versatility. It is frequently desirable to be able to highlight regions of structures, as done by kinemage, in these systems. While kinemages can be sent over the network and are part of the proposed chemical MIME type (Borenstein & Freed 1993; Rzepa, Murray-Rust & Whitaker 1995), they are not fully suitable for general use. In part, this is because the data needed to specify a region of a protein and its annotation are relatively small but coordinates—especially in the PDB format—take an enormous number of bytes. In the case of structures deposited in the PDB, the coordinate data in a kinemage file is no different from that which is already likely to exist locally. By conflating the commentary (specifying regions) with the coordinate data, kinemage files are made unnecessarily large and therefore inconvenient for sending over networks. Additionally, the structures in kinemage format cannot be viewed with display systems which may have facilities which differ from those in the kinemage programs. One partial solution is custom scripts for viewing systems, such as the rasmolscript system (Hubbard & Brenner 1994) which avoids mixing annotation with data but has the disadvantage of being specific to a particular program.

In recent years the number of programs for viewing and analyzing macromolecular structures have grown dramatically, as have individual programs' sophistication. It is remarkable that no standard mechanism currently exists for conveying information, beyond coordinates, from one system to another. Virtually every program has its own method of defining regions of proteins, and these specifications often have little overlap. It would clearly be desirable to have a format for inter-program communication. Unfortunately, the kinemage format is inappropriate for this role because it uses a non-standard format for macromolecular data, so the coordinate data cannot be easily extracted. Further, the highlighted regions lack the generality necessary for any sort of usage other than visual display.

The National Center for Biotechnology Information (NCBI) macromolecular database (MMDB) types (Ohkawa & Bryant 1994) and the Chemical Abstracts Service's Chemical eXchange Format (CXF) (Steckert & Mockus

1994) provide very powerful and general means of describing and annotating macromolecular structures, as does the proposed mmCIF format (Bourne *et al. in prep.*). However, these formats are complex to use and cannot be easily interpreted or created manually. Moreover, like the PDB format, they do not provide straightforward facilities for describing coordinates stored elsewhere or for specification of how a molecule should be displayed.

Thus, it is with the goal of providing program-independent annotation of macromolecular structures, without the need for integral coordinate information, that we propose the annmm set of standards. It is a simple, general, and easy to understand format which permits markup of PDB-format macromolecular structures similar to that available by use of the kinemage format. We expect that extensions and improvements will be made to the format in the future, though care has been taken to ensure the specification is robust enough to be backward compatible.

Design Goals

The key idea behind annmm is to refer to a set of macromolecular coordinates (generally the name of a PDB entry), and define a set of annotations, which can be display-independent markups. For a given type of highlighting, particular display techniques will be preferred, but the actual display system may choose any type of highlighting of which it is capable. For flexibility, nearly every one of the specification's objects may be omitted. This broadens the utility of the specification, but places a burden on systems reading the format to produce appropriate results when insufficient information is provided. The annmm specifications are very general, and can be used as a general-purpose format for exchanging data between different macromolecular databases, analysis programs, and display systems.

To ensure that it would be syntactically well-defined, annmm has been specified as a set of Abstract Syntax Notation 1 (ASN.1) objects. ASN.1 is a standard for representing data which has been adopted by the International Standards Organization (ISO) for open systems interconnection (OSI) in almost precisely the type of use intended for annmm. Its use for biological data was pioneered by NCBI, which maintains the GenBank and Entrez databases (Benson, Lipman & Ostell 1993). The Chemical Abstracts Service has also designed CXF, an extremely general way of describing small and large molecules, in ASN.1 (Steckert & Mockus 1994). A description of the ASN.1 notation, further references, and a more detailed rationale for its use in specifying biological data can be found in the NCBI Toolbox documentation (Ostell 1993).

While the ASN.1 notation provides strong syntactic definitions, it does not provide any guarantees about the semantic interpretation of an annmm object. The semantics have therefore been developed with careful attention to detail and completeness, although carefully selected portions of the annmm are semantically undefined or weakly defined. There are several reasons for this. First, it permits modifications to the format which take into account the experience of use, without the need to modify the syntax (and therefore render the data unreadable to older parsers). Second, it delegates some of the interpretation of the objects to the systems making use of them, allowing the systems to incorporate customized features. Because of the way in which the specification is designed, it is possible for individual systems to develop their own extensions within the defined syntax (i.e., without having to define new ASN.1 objects) without corrupting the data in such a way that it cannot be used by other systems. Third, the fuzzy definition of some semantics provides flexibility for human communication.

Wherever possible, the format has been designed to place burdens on the creator of annmm objects rather than the reader of them. As a result, when several redundant means of describing information could be used, annmm tries to permit only one, even though this occasionally makes the format somewhat inconvenient or obtuse. For example, there is no way to specify a region as "all of the threonines" in a protein; the actual residue numbers of each threonine would need to be provided. The rationale for this approach is to minimize the overhead required to use the annmm format; generally the specifications used by annmm are among those used by most programs or are at least easily derived from those used by most programs. The principal disadvantage of this minimalist approach is that human users may need to use tools to generate annmm objects for apparently simple specifications. But providing alternative means of specifying data would require every program wishing to use annmm objects to include additional code to translate a variety of formats into the preferred internal format.

Semantics of the Annmm Standard

Annmm Object

Typical users of the annmm standard will store and transfer Annmm objects (defined in Figure 1; example in Figure 2). These objects contain information about the macromolecular structure data being described, general viewing parameters, and a set of display elements which include the actual region-defining objects. It is effectively a container for holding the region information and specifying the context to which they refer.

The Annmm objects consist of a sequence of parameters, none of which need be provided save an identifier specifying that the object is of the Annmm type. A detailed explanation of each field follows:

- `type`
  For identification of the type of the Annmm object, and its version, these are provided as the first, and sole mandatory fields in the annmm object. The type listed here should be the proposed chemical MIME (Rzepa, Murray-Rust & Whitaker 1995) name for the object.
- `creator`
  The system which created the Annmm object may

identify itself here. No standard for identifying programs is prescribed. This information may be used to aid interpretation of the mark-up.

- `id`
  This identifier will be used in later versions of the annmm specification to uniquely identify set of data; presently it is unused.
- `title`
  The title is a simple string used for describing the Annmm object. It should provide all necessary context, and should not depend upon any other element of the object for explanation
- `comment`
  The comment field is intended to contain general information about the object, for human consumption. A typical comment would explain, in general terms, what the Annmm object is intending to show, and might be considered to be a caption to the object as a means of communications. Although typed comments [see below] are permitted here, generally machine-readable information would be provided in the custom field. Thus, beyond a possible single series, nested comments would not usually be expected here.
- `format`
  If the format is not pdb, this field may be used to select a reference database. The entry in the database is indicated by the name field.
- `name`
  The name is intended to hold the 4-character PDB entry code for the coordinate data, or an identifier for an entry an a different database specified by the format field. If no standard identifier exists, the name should be omitted.
- `location`
  Because the Annmm object does not contain integral coordinate data, it is necessary the systems using these objects know how to locate the data. Most standard macromolecular coordinate files can be found in well-known locations (either locally or over the internet), which can be deduced from the name of the entry and its format. Thus, in most cases the data will be retrieved on the basis of those two fields alone. The location field provides alternative location an to look for the data should the regular locations fail or should a name not be given. If a name is given, then the data provided by the location field is not guaranteed to be used. There are two ways of providing location information: either a URL (which can specify a file either on a local disk or over the network) or "here," which means that the actual coordinate data is incorporated directly into the Annmm object.
- `transform`
  Global transformations and viewing parameters to be applied to the coordinates are specified by a Transform object, described in more detail below.
- `custom`
  This field is intended for use by individual programs to provide data in a customized, implementation-dependent format. Typically the comment will consist of a series of typed comments. The type of each member of the series indicates what program is intended to make use of the data, and the structure of the comment associated with the type is left to the discretion of that program. Comments are described in greater detail below.
- `context`
  The context provides the facility to entirely mask the coordinate data described by the element regions. Specifying a context is tantamount to deleting all coordinate data which does not fall within the context. This may be used, for example, to indicate that all elements describe only one of several copies of a single protein in a crystal unit. It also provides a simple and coarse mechanism for focusing on a single region of a macromolecule.
- `elements`
  The elements field contains a series of AnnElement objects, which contain the actual regions of the structure to be annotated as well as the annotations and display features, as described below.

AnnElement Object

An AnnElement object contains a definition of a region of a macromolecule and annotation which most commonly will indicate how the region should be displayed. The coordinates to which the region applies are specified elsewhere, generally in the enclosing Annmm object. In the current version of the definition, the means of formally describing the region are limited: only a title (preferably very short), DisplayFormat objects, and comments are permitted. Both of the latter objects are described in greater detail below. While currently unused, the AnnElement can also include an identification key, which may be used by future versions of the annmm specification to permit AnnElements to be combined with each other in more complex ways than those currently possible.

AnnRegion Object

The AnnRegion object is the core of the annmm specification and is a single VisibleString with data encoded in a custom format (i.e., not ASN.1). Although ease of computer generation and parsing of AnnRegions is considered more important than manual interpretation, they are still intended to be easily comprehensible and usable by humans.

As noted above, the mechanism for describing regions is complete but avoids multiple mechanisms for describing the data. Thus, it may be less convenient than the languages provided by some programs for specifying certain characteristics of proteins. In these cases, tools may be used to generate the annmm object.

The AnnRegion object contains sets of many fields which each act to restrict the data in the region, and all of these fields are logically ANDed together to generate a final set of atoms. These sets may then be connected with a logical OR. These sets are specified by strings ORed with the vertical bar

(|) character. In most cases, a field of the definition not specified will select all possible values, while an empty string for a given field will select nothing. Within each field, the hyphen character (-) is an inclusive range operator and the comma character (,) acts as a logical OR. Spaces are not permitted, and all text within an item is case insensitive. Items listed individually or as a terminus of a range must exist in the coordinate data; however, there may be gaps within the sequence of a range.

Most of the field names in the specification (Figure 1) have an obvious relationship to items specified in the PDB format. Nonetheless, some explanation is required because there are several caveats. Moreover, it should be noted that many fields cannot be specified completely on a "general" PDB file; the vagaries of the particular PDB file must be used to determine precisely what will be selected.

- `model`
  If no model is selected, the system reading the file may choose to operate on only the first model rather than on all.
- `chain`
  A chain range may be only alphabetic or numeric and does not relate to the ordering of chains in the ATOM records of the PDB file. Thus the range "A-C" specifies only chains A, B, C even if the PDB chain order is A,L,B,1,2,M,C,N; to specify 1,2,A,B use "1-2,A-B."
- `mer`
  A 'mer' is a single monomer of a macromolecular polymer; thus, it is generally a protein's amino acid residue or a nucleic acid's nucleotide, the atoms of which are specified by a PDB ATOM record. When atoms with insertion codes (e.g., a residue with the name 38A) fall between residues specified by a range or are specifically listed in a range, then those residues are part of the range. A missing mer field will only select all mers if a het field is not defined, as explained below.
- `het`
  The het field is analogous to the mer field, except that it refers to "heterogens" whose atoms are given by HETATOM records in a PDB file. The annmm specification permits monomers and heterogens to have independent number spaces. Thus it is possible to have both a residue or base 3 and a heterogen 3 in a given chain, with the former being specified by the mer field, an the later by the het field. However, as mer and het numberings usually do not overlap, if one of mer or het is specified and the other is not, then the one not specified will be considered as a null selection (rather than no selection at all). Consequently, selecting mers 1-30 will preclude the selection of hets 31-40 unless they are explicitly specified.
- `alt`
  The PDB format permits a given atom to have partial occupancy in each of several alternative positions. The alt field allows a preference to be expressed for one or more of these positions. When only one position for an atom exists, then this field is ignored, as alternatives are usually only named in PDB files when more than one configuration is available. It should be noted that while alternative atom positions in a PDB file are specified on a per-atom level, it is usually only meaningful to construct residues (and in some cases, whole proteins) will all atoms in the same (e.g., "B") alternative configuration. If no alternative is explicitly listed, the system may decide whether to display a single alternative (generally the first or that with the highest occupancy) or all alternatives.
- `atomtype`
  Individual atoms within a residue are specified by their type, as defined by the PDB. The type includes both the element type of the atom and the atom's position within the monomer or heterogen. For example, the alpha carbon is called "CA." The PDB also specifies orderings for atom names, which may in the future be used in ranges, but are not currently permitted.

The spec definition consists of a single string. Each field provided must be in the appropriate order and in most cases must be preceded or suffixed by an identifying character as follows:

[model$][chain:][mer][#het][^alt][/atomtype]

If the proposed SegID field is added to PDB entries, then that will constitute another element of the spec string. It is expected that it would fall between chain and mer, and be suffixed by a percent sign.

As noted above, blocks of these specifications may be joined by vertical bars to OR them together. Some sample specifications are shown and explained below:

- 1-100
  Select all atoms in residues 1-100. If multiple chains are present, select those residues from all chains. The program may choose its own default actions to deal with multiple models and alternative configurations if they are present. No atoms from heterogens will be selected.
- A:48A^B
  Select conformation B (if such a conformation exists) of residue 48A in chain A.
- 20-50$A-C:20-40#1-10^A/N,O,C,CA,CB
  Select the backbone atoms and beta carbons in models 20 through 50, chains A, B, C, residues 20-40 (including inserted residues such as 32A), and using alternative A whenever multiple configurations exist and the oxygen ("O") atoms in waters (heterogens) 1-10.
- A,1:|/CA
  Select all atoms in the chains named A and 1, and all alpha carbons in the entire coordinate data.

Comment Object

Typically, a comment will simply be either a VisibleString of free text, or a URL which references relevant textual or other information. When multiple strings or URLs are needed, the series option can be used. Nested Comments are semantically undefined in general, but may be used in implementation-specific manner.

An important alternative use of the Comment object is to embed system dependent extensions within an annmm specification using the "typed" field choice. A Comment can

be identified as being intended for a particular program in the type field, and the comment field can provide text in an implementation-defined manner. The Comment type is recursive, and once the intended interpreting program is specified at the highest level, all other levels are left to that system to use in its desired manner. With experience, we expect that standard usage guidelines will evolve. A sample of a Comment providing type-specific information is in Figure 3. Typed Comments intended for custom use must be at the highest level available, either as the sole choice for the comment, or as one of the members of a series at the highest level.

Transform Object

The Transform object may be included in an Annmm object to provide global transformations of the provided coordinates, as noted in the description of the Annmm object. This permits a molecule specified by a standard file to be re-oriented to highlight specific features. Some transformations are order-dependent, and must be carried out in the order specified by the format. There are three components to the transformation: rotation, translation, and zoom. Each of these is specified as a choice because there are a number of different methods of specifying them. Presently, however, only one format is defined for each one.

The rotation is given as a set of rotation angles, in radians, around each of the x, y, and z axes in order. To provide the desired view, it is necessary that display system show x horizontally and positive to the right, y vertically and positive upwards and positive z projecting out of the screen. Translations are specified as a simple angstrom offset to be added to each point, and viewing programs are assumed to try to place xyz coordinates 0,0,0 at the center of the screen. (If no translation if given, then the system may center the coordinates as it wishes.) Zoom does not necessarily affect the coordinates themselves, but is typically used in viewing systems to highlight a particular region. It is normally expected that the display will encompass the whole of the macromolecule, and this is equivalent to a zoom magnification of 1. If a particular Annmm object wished to focus on a small site on a large protein, rotation could be used to bring the site to the fore, and translations to center the region, and then zoom to have it fill the whole screen.

In order to make the format easy to parse, and place computational burdens on systems generating the format rather than those reading it, the transformations do have some fairly inconvenient limitations. The most significant is that all operations must be based upon numerical coordinates, and not features of the macromolecule (i.e., it is not possible to automatically center on a particular residue). In addition, there is no explicit facility for clipping, although defined AnnRegions could be used to effectively provide this. A minor inconvenience is that many alternative means of describing the rotation, including matrices, quaternions, and rotation around a defined line are not currently supported.

DisplayFormat Object

Hints about the how a region of a macromolecule should be displayed are given in the DisplayFormat object. Presently, it only defines fairly abstract descriptions or system-defined "custom" descriptions in Comment format (above). The custom descriptions may use explicit commands from the viewer system or any other useful information.

The abstract display selection selected must be one of many enumerated formats, which fall into two general categories. The first provide some hints to the system about what is important about the selected region, for example, that it is an alpha helix. The display system could use this information to, for example, display the region as a helix rather than a set of bonds. However, the system is not obliged to use any particular representation.

The second category consists of entirely abstract descriptions, names only d00 through d09. Some suggestions for how a viewer may wish to interpret these are given as comments and there is an implied hierarchy of d01 being more significant than d02 (and so on to d09), but the only onus placed on the displayer is to attempt to present the ten different formats differently. Display systems may use the creator data from the Annmm object to help decide what display method to use for the fully abstract types. A minimal attempt to provide differentiation between the different forms of highlight might simply involve different colors or shadings.

Future Directions

Libraries to read and write the annmm objects in the C and Perl languages are currently under development, as are interfaces for scop (Murzin *et al.* 1995), RasMol (Sayle 1994) Molscript (Kraulis 1991), and Kinemage (Richardson & Richardson 1992). We hope that the broad utility of this standard for annotating regions for structures will lead to its widespread use. Increased use will, inexorably, highlight some weaknesses of the format and lead to requests for more flexibility and features. One of the challenges of maintaining the annmm specification will be deciding how to enhance it without making it inconveniently complex.

Many extensions to annmm are already planned, but are not included in the first version, both to speed adoption of the standard and to allow experience to guide the architecture. In order to facilitate comparison of molecules, some sort of container for Annmm objects will be created. It will probably permit references to symbolically named Annmm and AnnElement objects. Another set of new objects will be created to enhance the types of annotation available, such as to allow the definition of secondary structure elements. Many new display options will be added as they are deemed necessary, including animation and color, as well as more abstract types and the ability to define a few "explicit" types with parameters (e.g., alpha-carbon backbone 1 Å wide.)

It is perhaps ironic that a specification for regions of macromolecules stored in PDB format should created just as that format enters its twilight. While the annmm set of

objects will be able to specify most of the entities in newer formats such as mmCIF, MMDB and CXF, it will require some straightforward extensions to do so fully. Moreover, if most programs which deal with macromolecules incorporate a STAR-based parser to read mmCIF, it may be profitable to translate the annmm specification to STAR, even though ASN.1 does has broader support and some technical advantages. However, for at least the next several years, and probably longer, the PDB format will be used predominantly. During that time and probably thereafter, the annmm format will help make macromolecular structures a powerful and direct means of scientific communication.

Standards Note

We recommend that the Annmm type be registered as MIME type `chemical/annmm`, for file extensions `annmm` and `amm`. Until the chemical MIME primary type is adopted, we recommend that Annmm object be transmitted under the type `chemical/x-annmm`.

Acknowledgments

Drs. Phil Bourne, Stephen Bryant, Peter Murray-Rust, Henry Rzepa, Roger A. Sayle, and David Stampf helped refine the annmm standard.
S.E.B. is principally supported by a St. John's College (Cambridge) Benefactors' Scholarship. T.J.P.H. is supported by the MRC and ZENECA.

References

Abola, E.; Bernstein, F. C.; Bryant, S. H.; Koetzle, T. F.; and Weng, J. 1987. Protein Data Bank. In Allen, F. H.; Bergerhoff, G.; and Sievers, R., eds. *Crystallographic Databases - Information Content, Software Systems, Scientific Applications* Cambridge, UK: Data Commission of the International Union of Crystallography. pp. 107-132.

Benson, D.; Lipman, D. J.; and Ostell, J. 1993. GenBank. 21: 2963-2965.

Borenstein, N.; and Freed, M. 1993. MIME (multipurpose internet mail extensions) part one: Mechanisms for specifying and describing the format of internet message bodies. RFC 1521. (See, for example: `http://info.internet.isi.edu/in-notes/rfc/files/rfc1521.txt`)

Bourne, P.; *et al.,* The macromolecular crystallographic information file (mmCIF). *in preparation.*

FitzGerald, P. C. 1994. Molecules R Us. `http://www.nih.gov/htbin/pdb`

Hall, S. 1995. Protein images update natural history. *Science* 267:620-624.

Hubbard, T.; and Brenner, S. E. 1994. RasMol Scripts. `http://scop.mrc-lmb.cam.ac.uk/scop/rs/`

Kraulis, P. J. 1991. MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. *Journal of Applied Crystallography* 24:946-950.

Murzin, A. G.; Brenner, S. E.; Hubbard, T.; and Chothia, C. 1995. scop: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* 247:636-540. (See also: `http://scop.mrc-lmb.cam.ac.uk/scop/`).

Neurath, H. 1992. Why Protein Science? *Protein Science* 1:1–2.

Ohkawa, H.; and Bryant, S. H. 1994. The MMDB Specification. `ftp://ncbi.nlm.nih.gov/pub/mmdb`

Ostell, J. 1993. The NCBI Toolbox. `ftp://ncbi.nlm.nih.gov/toolbox`

Richardson, D. C.; and Richardson, J. S. 1992. The kinemage: A tool for scientific communication. *Protein Science* 1:3-9. (See also: `http://www.prosci.uci.edu/kinemages/kinpage.html`)

Rzepa, H. S.; Murray-Rust, P.; and Whitaker, B. J. 1995. A chemical primary content type for multipurpose internet mail extensions. `http://www.ch.ic.ac.uk/hypermail/chemime`

Sayle, R. 1994. RasMol. `ftp://ftp.dcs.ed.ac.uk/rasmol`

Skora, J. 1994. Protein Data Bank WWW server. `http://www.pdb.bnl.gov/`

Steckert, T.; and Mockus, J. 1994. Chemical eXchange Format (CXF). Chemical Abstracts Service `http://ibc.wustl.edu/standards`

White, S.; Falevsky, L; Frost L; Franklin S.; and Wiedeman, L. 1994. The Electronic *Protein Science*. `http://www.prosci.uci.edu/`

Figure 1. The Annmm ASN.1 Specification

```
Annmm DEFINITIONS ::=
BEGIN

EXPORTS Annmm;

Annmm ::= SEQUENCE {
    type        Type,               -- chemical/annmm, version 1.0
    creator     Type OPTIONAL,
    id          VisibleString OPTIONAL,     -- Currently unused; for unique id's
    title       VisibleString OPTIONAL,     -- user-readable title for display
    comment     Comment OPTIONAL,   -- comments
    format      ENUMERATED {
        pdb (1),
        other (255)
    } DEFAULT pdb,
    name        VisibleString OPTIONAL,     -- entry name, e.g., '2cro'
    location    CHOICE {                    -- where to find the entry
        url     VisibleString,              -- a URL (including local path)
        here    VisibleString               -- text of the actual object
    } OPTIONAL,
    transform   Transform OPTIONAL,
    custom      Comment OPTIONAL,   -- non-general items for recipient
    context     AnnRegion OPTIONAL,
    elements    SEQUENCE OF AnnElement OPTIONAL,
}

AnnElement ::= SEQUENCE {
    id      VisibleString OPTIONAL,     -- used for reference in higher objects
    title   VisibleString OPTIONAL,
    region  AnnRegion OPTIONAL,
    format  DisplayFormat OPTIONAL,
    comment Comment OPTIONAL,
}

AnnRegion ::= CHOICE {
-- These assume that residues and heterogens come from different numberspaces.
-- Lack of specification assumes "all", except that residue & hetresidues
--   exclude the other unless both are listed

        spec        VisibleString
    -- spec ::= [model$][chain:][residues][#hetresidues][^alternative][/atoms]
    }
}

Type ::= SEQUENCE {
    type    VisibleString,
    version VisibleString OPTIONAL
}

Comment ::= CHOICE {
    text    VisibleString,
    url     VisibleString,
    series  SEQUENCE OF Comment,
    typed   SEQUENCE {
        type    Type,
        comment Comment
```

```
    }
}

-- Default display should have coordinate axes with X horizontal, Y vertical,
-- Z positive out of screen.  0,0,0 should be at center of screen, and the
-- molecule should approximately fill the window, without clipping in any
-- direction.
Transform ::= SEQUENCE {
    rotation     CHOICE {
        axes     SEQUENCE {      -- rotate around each axis in order, in radians
            x    REAL,
            y    REAL,
            z    REAL } } OPTIONAL,
    translation CHOICE {
        offset SEQUENCE {       -- angstroms
            x    REAL,
            y    REAL,
            z    REAL } } OPTIONAL,
    zoom         CHOICE {
        mag      REAL } OPTIONAL -- magnification, 1 == default
}

DisplayFormat ::= SEQUENCE  {
    abstract     ENUMERATED {
        default (0),        -- some sort of default view
        general (20),       -- "general" display, e.g., CA backbone
        detailed (21),      -- detailed display e.g., wireframe
        atom (30),          -- some representation of atoms, e.g., colored points
        cpk (31),           -- spacefilling display
        bond (40),          -- some representation of bonds, e.g., wireframe
        hbond (41),         -- highlight hydrogen bonds
        atombond (50),      -- some representation with both atoms and bonds
        bas (51),           -- ball and stick
        secondary (60),     -- highlighting secondary structure
        alpha (61),
        beta (62),
        loop (63),
        turn (64),
        mer (70),           -- highlight the chemical nature of the monomers
        hydro (71),         -- according to hydrophobicity
            -- Highly symbolic types, to be interpreted, system-dependent
        d00 (100),          -- (default; wireframe color by atom type or solid)
        d01 (101),          -- (spacefilling)
        d02 (102),          -- (van der Waals surface)
        d03 (103),          -- (ball and stick)
        d04 (104),          -- (rods colored by mer class [e.g., hydrophobic])
        d05 (105),          -- (indicate secondary structure)
        d06 (106),          -- (wireframe color bright)
        d07 (107),          -- (wireframe color normal)
        d08 (108),          -- (de-emphasize; wireframe color dull)
        d09 (109)           -- (don't display at all)
    } OPTIONAL,
    custom      Comment OPTIONAL
}

END                             -- end of module
```

Figure 2. An example Annmm object

```
Annmm ::= {
    type {
        type        "chemical/x-annmm",
        version     "1.0" },
    creator {
        type        "scop",
        version     "0.5" },
    title       "scop: Globin fold in diphtheria toxin",
    name        "1mdt",
    location    url     "ftp://ftp.pdb.bnl.gov/...",
    context     spec    "A-B:",
    elements {
        {  region spec "A:",
            format {
                abstract    default },
            comment text "A whole protein" },
        {  region spec "B:",
            format {
                abstract    d09 },
            comment text "Duplicate subunit" },
        {   region spec "A:200-380",
            format {
                abstract    d03 },
            comment series {
                text    "The globin core of diphtheria toxin",
                url     "http://scop.mrc-lmb.cam.ac.uk/scop/..." } } } }
```

Figure 3. An example annmm Comment object

```
Comment ::=  {
    series {
        text    "A general-purpose comment for use by all systems.  Cannot be parsed.",
        url     "http://www.university.edu/relevant/url.html",
        typed {
            type    {
                type    "pdbviewer",
                version "5.0" },
            comment series {
                text    "background black",
                text    "depthcue on",
                text    "stereo off" }  },
        typed {
            type    {
                type    "pdbanaysis",
                version "0.5" },
            comment series {
                typed   {
                    type {
                        type    "prolog" },
                    comment text    "anisotropic cutoff" },
                typed   {
                    type {
                        type    "body" },
                    comment url     "http://www.u.edu/body-code.c" },
                types   {
                    type {
                        type    "coda" },
                    comment text    "output annmm" } } } } }
```